UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/679,870	10/06/2003	Patrick Williams	5150-77600	3836

7590 03/10/2008

Jeffrey C. Hood
Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767

EXAMINER	
CHEUNG, HUBERT G	

ART UNIT	PAPER NUMBER
2168	

MAIL DATE	DELIVERY MODE
03/10/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/679,870	WILLIAMS ET AL.	
	Examiner	Art Unit	
	Hubert Cheung	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *18 December 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-14,20-27,31 and 32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-14,20-27,31 and 32* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This Office action is in response to the amendment, arguments and remarks, filed on 12/18/2007, in which claims 1-14, 20-27, 31 and 32 are presented for further examination.

Claims 1, 7-12, 20, 21 and 23-27 have been amended.

Claims 15-19 and 28-30 have been cancelled.

Claims 31 and 32 have been added.

Response to Amendment

2. Applicant's arguments with respect to claims 1-14, 20-27, 31 and 32 have been considered but are moot in view of the new ground(s) of rejection, necessitated by applicant's amendment filed on 12/19/2007. Accordingly, this action has been made FINAL.

Applicant's amendment to the specification has been accepted. The objection to the use of trademarks has been withdrawn.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-14, 20-27 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Molinari, et al., US 2003/0058280 A1, filed on August 28, 2002 and published on March 27, 2003 (hereinafter "Molinari") in view of Bowman, et al., US 6,233,726 B1, filed on May 7, 1997 and issued on May 15, 2001 (hereinafter "Bowman").

Claim 1

With respect to claim 1, Molinari discloses **a computer-readable memory medium** (Molinari, paragraph [0083] where [c]omputer (12) comprises generally the internal components . . . that are typical of personal computers, including besides the CPU a main memory) **comprising program instructions executable to: dynamically determine a plurality of valid parameter values for the first parameter of the first function call** (Molinari, paragraph [0089] when queried by the user . . . the data sink presents to the user a detailed listing of available data sources; Molinari, paragraph [0148] where [u]pon the opening of the property page by the user, said property page handler browses front panel Aspect Handler . . . for any data source panels then existing within the front panel; and Molinari, paragraph [0150], where the list of available data channels is the plurality of valid parameter values for the first parameter of the first function call);

display a graphical user interface for selecting a parameter value for the first parameter of the first function call, wherein the graphical user interface for selecting the parameter value visually indicates the plurality of valid parameter

values (Molinari, paragraph [0089] where simply by clicking open the property page of the data sink "panel" placed on the desktop by the user[ ], the data sink presents to the user a detailed listing of available data sources; for any said data source selected by the user, said property page then displays a tree view of only those data channels; and Molinari, Fig. 13, where the list of available data channels is the plurality of valid parameter values for the first parameter of the first function call is displayed to the user);

receive user input to the graphical user interface to select a first parameter value from the plurality of valid parameter values (Molinari, paragraph [0089] where [u]pon selection of a desired data channel by the user); and

automatically include the first parameter value in the first function call in the source code of the software program in response to the user input selecting the first parameter value (Molinari, paragraph [0037] where [a]s the user places selected virtual instrument "panels" on the desktop and configures their properties . . ., an AIL file containing a description of the selected, created and defined aspects is simultaneously created, including, for each aspect, a description of its properties and connections [i.e., source code]), **wherein automatically including the first parameter value in the first function call aids a user in editing the first function call** (Molinari, paragraph [0032] where the execution of the developed application may therefore be implemented, with no compilation or interpretation of code, by the execution of a simple textual script file [i.e., source code] identifying the selected attributes, functions and connectivity determined by the user's application). However, Molinari is silent with respect to display source code of a software program, wherein the source code is

written in a text-based programming language that can be compiled into executable code, wherein the source code includes a first function call that takes a first parameter; wherein the graphical user interface for selecting the parameter value is displayed concurrently with the source code; and wherein automatically including the first parameter value in the first function call comprises automatically updating the displayed source code to display the first parameter value within the first function call. On the other hand, Bowman discloses **display source code of a software program, wherein the source code is written in a text-based programming language that can be compiled into executable code, wherein the source code includes a first function call that takes a first parameter** (Bowman, Col. 7, lines 15-28, where a code editor displays source code in C++, which is a text-based programming language that can be compiled into executable code);

wherein the graphical user interface for selecting the parameter value is displayed concurrently with the source code (Bowman, Col. 7, lines 15-18, where the C++ code is displayed);

wherein automatically including the first parameter value in the first function call comprises automatically updating the displayed source code to display the first parameter value within the first function call (Bowman, Col. 7, lines 15-18, wherein when changes to the C++ code is made they are reflected in the source code displayed in the code editor). It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate Bowman's teachings to Molinari's invention. A skilled artisan would have been motivated to do so, as

suggested by Bowman, Col. 2, line 66-Col. 3, line 7, in order to provide tools to facilitate creation and editing of source code. In addition, both of the references (Molinari and Bowman) disclose features that are directed to analogous art and they are directed to the same field of endeavor, such as creating software. This close relation between both of the references highly suggests an expectation of success.

Claim 2

With respect to claim 2, the combination of Molinari and Bowman further discloses **wherein said dynamically determining the plurality of valid parameter values comprises dynamically determining the plurality of valid parameter values based on a configuration of a computer system** (Molinari, paragraph [0089] where when queried by the user . . . the data sink presents to the user a detailed listing of available data sources; and Molinari, paragraph [0148] where [u]pon the opening of the property page by the user, said property page handler browses front panel Aspect Handler . . . for any data source panels then existing within the front panel).

Claim 3

With respect to claim 3, the combination of Molinari and Bowman further discloses **wherein said dynamically determining the plurality of valid parameter values based on the configuration of the computer system comprises dynamically determining the plurality of valid parameter values based on a hardware configuration of the computer system** (Molinari, paragraph [0089] when

queried by the user . . . the data sink presents to the user a detailed listing of available data sources; Molinari, paragraph [0148] where [u]pon the opening of the property page by the user, said property page handler browses front panel Aspect Handler . . . for any data source panels then existing within the front panel; Molinari, paragraph [0180] where the listing of data source types that are, or are to be, connected to the computer [which includes hardware] . . . and that are supported by appropriated device driver software [which inherently means there is hardware]; and Molinari, paragraph [0283] where listed data source types represent all of the various data source types then supported by the software of the invention, and thus usable to connect to a corresponding physical device).

Claim 4

With respect to claim 4, the combination of Molinari and Bowman further discloses **wherein said dynamically determining the plurality of valid parameter values based on the hardware configuration of the computer system comprises programmatically examining information regarding the hardware configuration of the computer system** (Molinari, paragraph [0180] where device driver software programmatically examines information regarding hardware; and Molinari, paragraph [0183] where the DAQ Data Source panel [i.e., software] allows a user to set up conventional DAQ hardware device . . . and to configure the subsystems of said DAQ device).

Claim 5

With respect to claim 5, the combination of Molinari and Bowman further discloses **wherein said dynamically determining the plurality of valid parameter values based on the hardware configuration of the computer system comprises programmatically querying software associated with one or more hardware devices coupled to the computer system** (Molinari, paragraph [0180] where device driver software programmatically examines information regarding hardware; Molinari, paragraph [0183] where the DAQ Data Source panel [i.e., software] allows a user to set up conventional DAQ hardware device . . . and to configure the subsystems of said DAQ device; and Molinari, paragraph [0283] where listed data source types represent all of the various data source types then supported by the software of the invention, and thus usable to connect to a corresponding physical device).

Claim 6

With respect to claim 6, the combination of Molinari and Bowman further discloses **wherein said dynamically determining the plurality of valid parameter values based on the configuration of the computer system comprises dynamically determining a first plurality of valid parameter values** (Molinari, paragraph [0089] when queried by the user . . . the data sink presents to the user a detailed listing of available data sources; and Molinari, paragraph [0148] where [u]pon the opening of the property page by the user, said property page handler browses front

panel Aspect Handler . . . for any data source panels then existing within the front panel);

wherein the program instructions are executable to dynamically determine a second plurality of valid parameter values based on the configuration of the computer system after the configuration of the computer system has been changed (Molinari, paragraph [0266] where once a data source device has been selected, a listing of subsystems associated with said device is presented to the user; and Molinari, paragraph [0291] where [u]pon selecting a particular data source input channel, . . ., the user is provided by the property page of the data source panel with numerous configuration options. . . . The ranges of parameter values presented to the user are set to limits established by the operating specification of the pertinent hardware device).

Claim 7

With respect to claim 7, the combination of Molinari and Bowman further discloses **wherein said dynamically determining the plurality of valid parameter values comprises dynamically determining one or more parameter values corresponding to hardware devices coupled to a computer system** (Molinari, paragraph [0291] where [u]pon selecting a particular data source input channel, . . ., the user is provided by the property page of the data source panel with numerous configuration options. . . . The ranges of parameter values presented to the user are set to limits established by the operating specification of the pertinent hardware device);

wherein the first parameter value corresponds to a first hardware device (Molinari, paragraph [0082] where one of the instruments coupled to computer (12), where an instrument is a hardware device; paragraph [0275] where the data acquisition hardware device is a DAQ board; and Molinari, Fig. 2 (16) where a serial instrument is a hardware device);

wherein said automatically including the first parameter value in the first function call comprises automatically configuring the first function call with a reference to the first hardware device (Molinari, paragraph [0037] where [a]s the user places selected virtual instrument "panels" [i.e., first hardware device] on the desktop and configures their properties . . ., an AIL file containing a description of the selected, created and defined aspects is simultaneously created, including, for each aspect, a description of its properties and connections [i.e., source code], where the "panels" have pre-configured properties).

Claim 8

With respect to claim 8, the combination of Molinari and Bowman further discloses **wherein said dynamically determining the plurality of valid parameter values comprises dynamically determining one or more parameter values corresponding to resources of one or more hardware devices** (Molinari, paragraph [0291] where [u]pon selecting a particular data source input channel, . . ., the user is provided by the property page of the data source panel with numerous configuration

options. . . . The ranges of parameter values presented to the user are set to limits established by the operating specification of the pertinent hardware device);

wherein the first parameter value corresponds to a first resource of a first hardware device (Molinari, paragraph [0275] where the DAQ Controller property page includes a Data Sources drop-down list . . [t]he user's selection of a data source device on this drop-down list effects a connection between said data source device and the DAQ Controller panel);

wherein said automatically including the first parameter value in the first function call comprises automatically configuring the first function call with a reference to the first resource of the first hardware device (Molinari, paragraph [0037] where [a]s the user places selected virtual instrument "panels" [i.e., first resource of the first hardware device] on the desktop and configures their properties . . ., an AIL file containing a description of the selected, created and defined aspects is simultaneously created, including, for each aspect, a description of its properties and connections [i.e., source code], where the "panels" have pre-configured properties).

Claim 9

With respect to claim 9, the combination of Molinari and Bowman further discloses **wherein said dynamically determining the plurality of valid parameter values comprises dynamically determining** (Molinari, paragraph [0089] when queried by the user . . . the data sink presents to the user a detailed listing of available data sources; and Molinari, paragraph [0148] where [u]pon the opening of the property page

by the user, said property page handler browses front panel Aspect Handler . . . for any data source panels then existing within the front panel) **one or more GPIB resources** (Molinari, paragraph [0082] where Molinari discloses using DAQ resources, which are used in test equipment, through an I/O slot in a computer by using a PCI bus, ISA bus or an EISA bus. Molinari also discloses providing users with a system that is specifically adapted for the development of customized measurement environments for automated test equipment (Molinari, paragraphs [0073]-[0074]). These implications disclose, "one or more GPIB resources.");

wherein the first parameter value comprises a first GPIB resource (Molinari, paragraph [0082] where Molinari discloses using DAQ resources, which are used in test equipment, through an I/O slot in a computer by using a PCI bus, ISA bus or an EISA bus. Molinari also discloses providing users with a system that is specifically adapted for the development of customized measurement environments for automated test equipment (Molinari, paragraphs [0073]-[0074]). These implications disclose, "wherein the first parameter value comprises a first GPIB resource.");

wherein said automatically including the first parameter value in the first function call comprises automatically the first function call with a reference to the first GPIB resource (Molinari, paragraph [0037] where [a]s the user places selected virtual instrument "panels" on the desktop and configures their properties . . ., an AIL file containing a description of the selected, created and defined aspects is simultaneously created, including, for each aspect, a description of its properties and connections [i.e., source code], where the "panels" have pre-configured properties).

Claim 10

With respect to claim 10, the combination of Molinari and Bowman further discloses **wherein said dynamically determining the plurality of valid parameter values comprises dynamically determining** (Molinari, paragraph [0089] when queried by the user . . . the data sink presents to the user a detailed listing of available data sources; and Molinari, paragraph [0148] where [u]pon the opening of the property page by the user, said property page handler browses front panel Aspect Handler . . . for any data source panels then existing within the front panel) **one or more Visa resources** (Molinari, paragraph [0082] where Molinari discloses using DAQ resources, which are used in test equipment, through an I/O slot in a computer by using a PCI bus, ISA bus or an EISA bus. Molinari also discloses providing users with a graphical system that is specifically adapted for the development of customized measurement environments for automated test equipment (Molinari, paragraphs [0073]-[0074]). These implications disclose, "one or more Visa resources.");

wherein the first parameter value comprises a first Visa resource (Molinari, paragraph [0082] where Molinari discloses using DAQ resources, which are used in test equipment, through an I/O slot in a computer by using a PCI bus, ISA bus or an EISA bus. Molinari also discloses providing users with a graphical system that is specifically adapted for the development of customized measurement environments for automated test equipment (Molinari, paragraphs [0073]-[0074]). These implications disclose, "wherein the first parameter value comprises a first Visa resource.");

wherein said automatically including the first parameter value in the first function call comprises automatically configuring the first function call with a reference to the first Visa resource (Molinari, paragraph [0037] where [a]s the user places selected virtual instrument "panels" on the desktop and configures their properties . . ., an AIL file containing a description of the selected, created and defined aspects is simultaneously created, including, for each aspect, a description of its properties and connections [i.e., source code], where the "panels" have pre-configured properties).

Claim 11

With respect to claim 11, the combination of Molinari and Bowman further discloses **wherein said dynamically determining the plurality of valid parameter values comprises dynamically determining one or more DAQ resources** (Molinari, paragraph [0291] where [u]pon selecting a particular data source input channel, . . ., the user is provided by the property page of the data source panel with numerous configuration options. . . . The ranges of parameter values presented to the user are set to limits established by the operating specification of the pertinent hardware device [which can include a DAQ resource]);

wherein the first parameter value comprises a first DAQ resource (Molinari, paragraph [0183] where [t]he DAQ Data Source panel allows a user to set up a conventional DAQ hardware device);

wherein said automatically including the first parameter value in the first function call comprises automatically configuring the first function call with a reference to the first DAQ resource (Molinari, paragraph [0037] where [a]s the user places selected virtual instrument "panels" [i.e., first DAQ resource] on the desktop and configures their properties . . ., an AIL file containing a description of the selected, created and defined aspects is simultaneously created, including, for each aspect, a description of its properties and connections [i.e., source code], where the "panels" have pre-configured properties).

Claim 12

With respect to claim 12, the combination of Molinari and Bowman further discloses **wherein said dynamically determining the plurality of valid parameter values comprises dynamically determining one or more universal resource locators (URLs)** (Molinari, paragraph [0082] where a data source can be obtained from another computer (20) acting as an OPC server. The OPC server can communicate with computer (12) via a wired or wireless means [i.e., network]. In order for computer (12) to communicate with computer (20), computer (12) would inherently have to dynamically determine computer (20)'s IP address. A URL is simply a pneumonic for an IP address. Therefore, the invention must inherently be able to dynamically determine one or more IP addresses, which could be URLs);

wherein the first parameter value comprises a first URL (Molinari, paragraph [0082] where a data source can be obtained from another computer (20) acting as an

OPC server. The OPC server can communicate with computer (12) via a wired or wireless means [i.e., network]. In order for computer (12) to communicate with computer (20), computer (12) would inherently have to dynamically determine computer (20)'s IP address. A URL is simply a pneumonic for an IP address. Therefore, the invention must inherently be able to have the first parameter value comprise a first URL to reach the data source on computer (20));

wherein said automatically including the first parameter value in the first function call comprises automatically configuring the first function call with a reference to the first URL (Molinari, paragraph [0037] where [a]s the user places selected virtual instrument "panels" [i.e., first URL] on the desktop and configures their properties . . ., an AIL file containing a description of the selected, created and defined aspects is simultaneously created, including, for each aspect, a description of its properties and connections [i.e., source code], where the "panels" have pre-configured properties).

Claim 13

With respect to claim 13, the combination of Molinari and Bowman further discloses **further comprising program instructions executable to: receive user input specifying filtering criteria for the parameter values** (Molinari, paragraph [0149] where [d]ata sources that fit the needs of data sink (130) are checked, and the resulting list of qualifying data sources is displayed, on a combo box, ... Upon the selection of a data source by the user; and Molinari, Fig. 10);

wherein the graphical user interface visually indicates only a subset of the valid parameter values, wherein the subset is determined based on the specified filtering criteria (Molinari, paragraph [0266] where once a data source device has been selected, a listing of subsystems associated with said device is presented to the user; Molinari, paragraph [0291] where [u]pon selecting a particular data source input channel . . . the user is provided by the property page of the data source panel with numerous configuration options. . . . The ranges of parameter values presented to the user are set to limits established by the operating specification of the pertinent hardware device; and Molinari, Fig. 10).

Claim 14

With respect to claim 14, the combination of Molinari and Bowman further discloses **further comprising program instructions executable to: receive user input requesting to display the graphical user interface for selecting the parameter value** (Molinari, paragraph [0149] where [a] list of panels having a data source aspect is returned, and an aspect filter (135) of data sink aspect (130) determines, for each data source found, whether the data source accords with the needs of said data sink aspect . . . . Data sources that fit the needs of data sink (130) are checked, and the resulting list of qualifying data sources is displayed, on a combo box, in a “data sources view” of the data sink property page; and Molinari, paragraph [0266] where once a data source device has been selected, a listing of subsystems associated with said device is presented to the user);

wherein said displaying the graphical user interface is performed in response to the user input requesting to display the graphical user interface (Molinari, paragraph [0266] where once a data source device has been selected, a listing of subsystems associated with said device is presented to the user).

Claim 20

With respect to claim 20, the combination of Molinari and Bowman further discloses

wherein the source code is displayed in a first window (Bowman, Col. 6, line 14-Col. 7, line 9, where a Classes window (371), Objects window (373), Files window (375) and Resource window (381) are described; and Bowman, Col. 7, lines 15-28, where a Code Editor (400) "window" is described as among the windows);

wherein said displaying the graphical user interface comprises displaying the graphical user interface in a separate window apart from the first window (Molinari, paragraph [0025] where [b]y the selections from menu lists, or the "drag and drop" of selected panels icons presented in a "flying tool windows" [which is apart from the main desktop and which is also a part of the graphical user interface of the software program]).

Claim 21

With respect to claim 21, the combination of Molinari and Bowman further discloses

wherein the source code is displayed in a first portion of a first window (Bowman, Col. 6, line 14-Col. 7, line 9, where a Classes window (371), Objects window (373), Files window (375) and Resource window (381) are described; and Bowman, Col. 7, lines 15-28, where a Code Editor (400) "window" is described as among the windows);

wherein said displaying the graphical user interface comprises displaying the graphical user interface in a second portion of the first window (Molinari, Fig. 1; Molinari, Fig. 16; and Molinari, Fig. 17; and Bowman, Fig. 7C, where there is graphical user interface displayed in the top [i.e., second] portion of the window as well as C++ code).

Claim 22

With respect to claim 22, the combination of Molinari and Bowman further discloses **wherein the graphical user interface displays the plurality of valid parameter values as a list** (Molinari, paragraph [0089] where [w]hen queried by the user (simply by clicking open the property page of the data sink "panel" placed on the desktop by the user), the data sink presents to the user a detailed listing of available data sources; for any said data source selected by the user, said property page then displays a tree view of only those data channels that are supported for display by the data sink panel; Molinari, Fig. 11; Molinari, Fig. 13; and Molinari, Fig. 15);

wherein said receiving user input to the graphical user interface to select the first parameter value comprises receiving user input to the graphical user

interface to select the first parameter value from the list (Molinari, paragraph [0089] where [w]hen queried by the user (simply by clicking open the property page of the data sink "panel" placed on the desktop by the user), the data sink presents to the user a detailed listing of available data sources; for any said data source selected by the user, ... Upon selection of a desired data channel by the user, the data sink aspect contains the functionality required to establish automatically a data link between the data source and the data sink).

Claim 23

With respect to claim 23, the combination of Molinari and Bowman further discloses **wherein said dynamically determining the plurality of valid parameter values includes dynamically determining one or more property values** (Molinari, paragraph [0089] where [w]hen queried by the user (simply by clicking open the property page of the data sink "panel" placed on the desktop by the user), the data sink presents to the user a detailed listing of available data sources; for any said data source selected by the user, said property page then displays a tree view of only those data channels that are supported for display by the data sink panel; and Molinari, paragraph [0149] where [a] list of panels having a data source aspect is returned, and an aspect filter (135) of data sink aspect (130) determines, for each data source found [i.e., dynamically determining a property], whether the data source accords with the needs of said data sink aspect . . . . Data sources that fit the needs of data sink (130) are

checked, and the resulting list of qualifying data sources is displayed [i.e., dynamically determining another property]);

wherein said receiving user input to the graphical user interface to select the first parameter value comprises receiving user input to the graphical user interface to select a first property value (Molinari, paragraph [0089] where [w]hen queried by the user (simply by clicking open the property page of the data sink "panel" placed on the desktop by the user), the data sink presents to the user a detailed listing of available data sources);

wherein the first property value is automatically included in the first function call in response to the user input selecting the first property value (Molinari, paragraph [0037] where [a]s the user places selected virtual instrument "panels" on the desktop and configures their properties . . ., an AIL file containing a description of the selected, created and defined aspects is simultaneously created, including, for each aspect, a description of its properties and connections [i.e., source code]; and Molinari, paragraph [0099] where [t]he actions of the user in selecting instrument component panels, and configuring their properties, serve to define the content and properties of the user's intended application program. These actions by the user also operate to create a textual file, called in this description an Aspect Interaction Language (AIL) file, that contains a description of the selected, created and defined software "aspects", ... As the user adds more panels, selects more properties, and effects connections between panels, the AIL file expands into a complete textual description of the user's program, where the "panels" have pre-configured properties).

Claim 24

With respect to claim 24, Molinari discloses **a computer-readable memory medium comprising program instructions executable to: determine a plurality of parameter values for the first parameter of the first method call based on a hardware configuration of a computer system** (Molinari, paragraph [0089] when queried by the user . . . the data sink presents to the user a detailed listing of available data sources; Molinari, paragraph [0148] where [u]pon the opening of the property page by the user, said property page handler browses front panel Aspect Handler . . . for any data source panels then existing within the front panel; and Molinari, paragraph [0150], where the list of available data channels is the plurality of valid parameter values for the first parameter of the first function call; and Molinari, paragraph [0291] where [t]he ranges of parameter values presented to the user are set to limits established by the operating specifications of the pertinent hardware device);

display a graphical user interface for selecting a parameter value for the first parameter of the first method call, wherein the graphical user interface for selecting the parameter value visually indicates the plurality of parameter values (Molinari, paragraph [0089] where simply by clicking open the property page of the data sink "panel" placed on the desktop by the user[ ], the data sink presents to the user a detailed listing of available data sources; for any said data source selected by the user, said property page then displays a tree view of only those data channels; and Molinari,

Fig. 13, where the list of available data channels is the plurality of valid parameter values for the first parameter of the first function call is displayed to the user);

receive user input to the graphical user interface to select a first parameter value from the plurality of parameter values (Molinari, paragraph [0089] where [w]hen queried by the user (simply by clicking open the property page of the data sink "panel" placed on the desktop by the user), the data sink presents to the user a detailed listing of available data sources); and

automatically include the first parameter value in the first method call in the source code of the software program in response to the user input selecting the first parameter value (Molinari, paragraph [0037] where [a]s the user places selected virtual instrument "panels" on the desktop and configures their properties . . ., an AIL file containing a description of the selected, created and defined aspects is simultaneously created, including, for each aspect, a description of its properties and connections [i.e., source code]; and Molinari, paragraph [0099] where [t]he actions of the user in selecting instrument component panels, and configuring their properties, serve to define the content and properties of the user's intended application program. These actions by the user also operate to create a textual file, called in this description an Aspect Interaction Language (AIL) file, that contains a description of the selected, created and defined software "aspects", ... As the user adds more panels, selects more properties, and effects connections between panels, the AIL file expands into a complete textual description of the user's program, where the "panels" have pre-configured properties), **wherein automatically including the first parameter value in the first method call**

aids a user in editing the first method call (Molinari, paragraph [0032] where the execution of the developed application may therefore be implemented, with no compilation or interpretation of code, by the execution of a simple textual script file [i.e., source code] identifying the selected attributes, functions and connectivity determined by the user's application). However, Molinari is silent with respect to **display source code of a software program, wherein the source code is written in a text-based programming language that can be compiled into executable code, wherein the source code includes a first method call that takes a first parameter**; wherein the graphical user interface for selecting the parameter value is displayed concurrently with the source code; and wherein automatically including the first parameter value in the first method call comprises automatically updating the displayed source code to display the first parameter value within the first method call. On the other hand, Bowman discloses **display source code of a software program, wherein the source code is written in a text-based programming language that can be compiled into executable code, wherein the source code includes a first function call that takes a first parameter** (Bowman, Col. 7, lines 15-28, where a code editor displays source code in C++, which is a text-based programming language that can be compiled into executable code);

wherein the graphical user interface for selecting the parameter value is displayed concurrently with the source code (Bowman, Col. 7, lines 15-18, where the C++ code is displayed);

wherein automatically including the first parameter value in the first method call comprises automatically updating the displayed source code to display the first parameter value within the first method call (Bowman, Col. 7, lines 15-18, wherein when changes to the C++ code is made they are reflected in the source code displayed in the code editor). See claim 1 above for the motivation to combine.

Claim 25

With respect to claim 25, Molinari discloses **a computer-readable memory medium comprising program instructions executable to: determine a plurality of resources of one or more measurement devices coupled to a computer system** (Molinari, paragraph [0082] where [t]he system (10) comprises a computer (12), which is connectable to a plurality of instruments [i.e., measurement devices]); Molinari, paragraph [0089] when queried by the user . . . the data sink presents to the user a detailed listing of available data sources; and Molinari, paragraph [0148] where [u]pon the opening of the property page by the user, said property page handler browses front panel Aspect Handler . . . for any data source panels then existing within the front panel; and Molinari, paragraph [0150], where the list of available data channels is the plurality of valid parameter values for the first parameter of the first function call);

display a graphical user interface visually indicating a plurality of parameter values for the first parameter of the first function call, wherein each parameter value corresponds to one of the resources (Molinari, paragraph [0089] where [w]hen queried by the user (simply by clicking open the property page of the data

sink "panel" placed on the desktop by the user), the data sink presents to the user a detailed listing of available data sources [i.e., graphical user interface]; and Molinari, Fig. 13, where the list of available data channels is the plurality of valid parameter values for the first parameter of the first function call is displayed to the user);

receive user input to the graphical user interface to select a first parameter value from the plurality of parameter values (Molinari, paragraph [0089] where [w]hen queried by the user (simply by clicking open the property page of the data sink "panel" placed on the desktop by the user), the data sink presents to the user a detailed listing of available data sources); and

automatically include the first parameter value in the first function call in the source code of the software program in response to the user input selecting the first parameter value (Molinari, paragraph [0037] where [a]s the user places selected virtual instrument "panels" on the desktop and configures their properties . . ., an AIL file containing a description of the selected, created and defined aspects is simultaneously created, including, for each aspect, a description of its properties and connections [i.e., source code]), **wherein automatically including the first parameter value in the first function call aids a user in editing the first function call** (Molinari, paragraph [0032] where the execution of the developed application may therefore be implemented, with no compilation or interpretation of code, by the execution of a simple textual script file [i.e., source code] identifying the selected attributes, functions and connectivity determined by the user's application). However, Molinari is silent with respect to display source code of a software program, wherein the source code is

written in a text-based programming language that can be compiled into executable code, wherein the source code includes a first function call that takes a first parameter; wherein the graphical user interface is displayed concurrently with the source code; and wherein automatically including the first parameter value in the first function call comprises automatically updating the displayed source code to display the first parameter value within the first function call. On the other hand, Bowman discloses **display source code of a software program, wherein the source code is written in a text-based programming language that can be compiled into executable code, wherein the source code includes a first function call that takes a first parameter** (Bowman, Col. 7, lines 15-28, where a code editor displays source code in C++, which is a text-based programming language that can be compiled into executable code);

wherein the graphical user interface is displayed concurrently with the source code (Bowman, Col. 7, lines 15-18, where the C++ code is displayed);

wherein automatically including the first parameter value in the first function call comprises automatically updating the displayed source code to display the first parameter value within the first function call (Bowman, Col. 7, lines 15-18, wherein when changes to the C++ code is made they are reflected in the source code displayed in the code editor). See claim 1 above for the motivation to combine.

Claim 26

With respect to claim 26, Molinari discloses **a system comprising: a processor** (Molinari, paragraph [0083] where [a]ccordingly computer (12) includes at least one central processing unit, or CPU . . . .);

a memory coupled to the processor, wherein the memory stores program instructions (Molinari, paragraph [0083] where [c]omputer (12) comprises generally the internal components . . . that are typical of personal computers, including besides the CPU a main memory);

wherein the program instructions stored in the memory are executable by the processor to: dynamically determine a plurality of valid parameter values for the first parameter of the first function call (Molinari, paragraph [0089] when queried by the user . . . the data sink presents to the user a detailed listing of available data sources; and Molinari, paragraph [0148] where [u]pon the opening of the property page by the user, said property page handler browses front panel Aspect Handler . . . for any data source panels then existing within the front panel; and Molinari, paragraph [0150], where the list of available data channels is the plurality of valid parameter values for the first parameter of the first function call);

display a graphical user interface for selecting a parameter value for the first parameter of the first function call, wherein the graphical user interface for selecting the parameter value visually indicates the plurality of valid parameter values (Molinari, paragraph [0089] where simply by clicking open the property page of the data sink “panel” placed on the desktop by the user[ ], the data sink presents to the user a detailed listing of available data sources; for any said data source selected by the

user, said property page then displays a tree view of only those data channels; and Molinari, Fig. 13, where the list of available data channels is the plurality of valid parameter values for the first parameter of the first function call is displayed to the user);

receive user input to the graphical user interface to select a first parameter value from the plurality of valid parameter values (Molinari, paragraph [0089] where [w]hen queried by the user (simply by clicking open the property page of the data sink "panel" placed on the desktop by the user), the data sink presents to the user a detailed listing of available data sources); and

automatically include the first parameter value in the first function call in the source code of the software program in response to the user input selecting the first parameter value (Molinari, paragraph [0037] where [a]s the user places selected virtual instrument "panels" on the desktop and configures their properties . . ., an AIL file containing a description of the selected, created and defined aspects is simultaneously created, including, for each aspect, a description of its properties and connections [i.e., source code]), **wherein automatically including the first parameter value in the first function call aids a user in editing the first function call** (Molinari, paragraph [0032] where the execution of the developed application may therefore be implemented, with no compilation or interpretation of code, by the execution of a simple textual script file [i.e., source code] identifying the selected attributes, functions and connectivity determined by the user's application). However, Molinari is silent with respect to display source code of a software program, wherein the source code is written in a text-based programming language that can be compiled into executable

code, wherein the source code includes a first function call that takes a first parameter; wherein the graphical user interface for selecting the parameter value is displayed concurrently with the source code; and wherein automatically including the first parameter value in the first function call comprises automatically updating the displayed source code to display the first parameter value within the first function call. On the other hand, Bowman discloses **display source code of a software program, wherein the source code is written in a text-based programming language that can be compiled into executable code, wherein the source code includes a first function call that takes a first parameter** (Bowman, Col. 7, lines 15-28, where a code editor displays source code in C++, which is a text-based programming language that can be compiled into executable code);

wherein the graphical user interface for selecting the parameter value is displayed concurrently with the source code (Bowman, Col. 7, lines 15-18, where the C++ code is displayed);

wherein automatically including the first parameter value in the first function call comprises automatically updating the displayed source code to display the first parameter value within the first function call (Bowman, Col. 7, lines 15-18, wherein when changes to the C++ code is made they are reflected in the source code displayed in the code editor). See claim 1 above for the motivation to combine.

Claim 27

With respect to claim 27, Molinari discloses **a method for modifying source code of a software program, the method comprising: dynamically determining a plurality of valid parameter values for the first parameter of the first function call** (Molinari, paragraph [0089] when queried by the user . . . the data sink presents to the user a detailed listing of available data sources; and Molinari, paragraph [0148] where [u]pon the opening of the property page by the user, said property page handler browses front panel Aspect Handler . . . for any data source panels then existing within the front panel; and Molinari, paragraph [0150], where the list of available data channels is the plurality of valid parameter values for the first parameter of the first function call);

displaying a graphical user interface for selecting a parameter value for the first parameter of the first function call, wherein the graphical user interface for selecting the parameter value visually indicates the plurality of valid parameter values (Molinari, paragraph [0089] where simply by clicking open the property page of the data sink "panel" placed on the desktop by the user[ ], the data sink presents to the user a detailed listing of available data sources; for any said data source selected by the user, said property page then displays a tree view of only those data channels; and Molinari, Fig. 13, where the list of available data channels is the plurality of valid parameter values for the first parameter of the first function call is displayed to the user);

receiving user input to the graphical user interface to select a first parameter value from the plurality of valid parameter values (Molinari, paragraph [0089] where [w]hen queried by the user (simply by clicking open the property page of

the data sink "panel" placed on the desktop by the user), the data sink presents to the user a detailed listing of available data sources); and

automatically include the first parameter value in the first function call in the source code of the software program in response to the user input selecting the first parameter value (Molinari, paragraph [0037] where [a]s the user places selected virtual instrument "panels" on the desktop and configures their properties . . ., an AIL file containing a description of the selected, created and defined aspects is simultaneously created, including, for each aspect, a description of its properties and connections [i.e., source code]), **wherein automatically including the first parameter value in the first function call aids a user in editing the first function call** (Molinari, paragraph [0032] where the execution of the developed application may therefore be implemented, with no compilation or interpretation of code, by the execution of a simple textual script file [i.e., source code] identifying the selected attributes, functions and connectivity determined by the user's application). However, Molinari is silent with respect to display source code of a software program, wherein the source code is written in a text-based programming language that can be compiled into executable code, wherein the source code includes a first function call that takes a first parameter; wherein the graphical user interface for selecting the parameter value is displayed concurrently with the source code; and wherein automatically including the first parameter value in the first function call comprises automatically updating the displayed source code to display the first parameter value within the first function call. On the other hand, Bowman discloses **display source code of a software program, wherein**

the source code is written in a text-based programming language that can be compiled into executable code, wherein the source code includes a first function call that takes a first parameter (Bowman, Col. 7, lines 15-28, where a code editor displays source code in C++, which is a text-based programming language that can be compiled into executable code);

wherein the graphical user interface for selecting the parameter value is displayed concurrently with the source code (Bowman, Col. 7, lines 15-18, where the C++ code is displayed);

wherein automatically including the first parameter value in the first function call comprises automatically updating the displayed source code to display the first parameter value within the first function call (Bowman, Col. 7, lines 15-18, wherein when changes to the C++ code is made they are reflected in the source code displayed in the code editor). See claim 1 above for the motivation to combine.

Claim 31

With respect to claim 31, Molinari discloses **a computer-readable memory medium comprising program instructions executable to: display a block diagram of a graphical program** (Molinari, Fig. 1, where a diagram of a “spectrum analyser" is depicted, where the blocks are “DT data source” (42), “DAQ controller” (44) and “Spectrum analyser” (46)),

wherein the block diagram includes a plurality of interconnected nodes visually indicating functionality of the graphical program (Molinari, Fig. 1, where

the "DT data source" (42) and the "DAQ controller" (44) are parts of the Spectrum analyser (46)),

wherein the block diagram can be compiled into executable code, wherein the plurality of interconnected nodes includes a first node that takes a first input parameter (Molinari, paragraph [0037] where [a]s the user places selected virtual instrument "panels" on the desktop and configures their properties . . ., an AIL file containing a description of the selected, created and defined aspects is simultaneously created, including, for each aspect, a description of its properties and connections [i.e., source code]; and Molinari, paragraph [0038], where the AIL file corresponds to an executable set of selections from existing libraries of executable code. This AIL file, which is human-readable, and readily saved and stored, also constitutes an abstract representation of the existing, executable code segments that are invoked, ordered and activated upon running the user's program");

dynamically determine a plurality of valid parameter values for the first input parameter of the first node (Molinari, paragraph [0089] when queried by the user . . . the data sink presents to the user a detailed listing of available data sources; Molinari, paragraph [0148] where [u]pon the opening of the property page by the user, said property page handler browses front panel Aspect Handler . . . for any data source panels then existing within the front panel; and Molinari, paragraph [0150], where the list of available data channels is the plurality of valid parameter values for the first input parameter of the first node);

display a graphical user interface for selecting a parameter value for the first input parameter of the first node (Molinari, paragraph [0089] where simply by clicking open the property page of the data sink "panel" placed on the desktop by the user[ ], the data sink presents to the user a detailed listing of available data sources; for any said data source selected by the user, said property page then displays a tree view of only those data channels; and Molinari, Fig. 13, where the list of available data channels is the plurality of valid parameter values for the first parameter of the first function call is displayed to the user), **wherein the graphical user interface for selecting the parameter value visually indicates the plurality of valid parameter values** (Molinari, paragraph [0150], where the list of available data channels is the plurality of valid parameter values for the first input parameter of the first node),

receive user input to the graphical user interface to select a first parameter value from the plurality of valid parameter values (Molinari, paragraph [0089] where [u]pon selection of a desired data channel by the user); and

automatically configure the first node with the first parameter value in response to the user input selecting the first parameter value (Molinari, paragraph [0037] where [a]s the user places selected virtual instrument "panels" on the desktop and configures their properties . . ., an AIL file containing a description of the selected, created and defined aspects is simultaneously created, including, for each aspect, a description of its properties and connections [i.e., source code]). However, Molinari is silent with respect to wherein the graphical user interface for selecting the parameter value is displayed concurrently with the block diagram; wherein automatically

configuring the first node with the first parameter value comprises automatically updating the displayed block diagram to visually indicate that the first node receives the first parameter value as input. On the other hand, Bowman discloses **wherein the graphical user interface for selecting the parameter value is displayed concurrently with the block diagram** (Bowman, Col. 7, lines 15-18, where the C++ code is displayed);

wherein automatically configuring the first node with the first parameter value comprises automatically updating the displayed block diagram to visually indicate that the first node receives the first parameter value as input (Bowman, Col. 7, lines 15-18, wherein when changes to the C++ code is made they are reflected in the source code displayed in the code editor). See claim 1 above for the motivation to combine.

5. Claim 32 is rejected under 35 U.S.C. 103(a) as being unpatentable over Molinari in view of Bowman and in further view of Shah, et al., US 2004/0032429 A1, filed on December 23, 2002 and published on February 19, 2004 (hereinafter "Shah").

Claim 32

With respect to claim 31, the combination of Molinari and Bowman further discloses **wherein automatically configuring the first node with the first parameter value comprises automatically wiring the first parameter value to an input terminal of the first node** (Molinari, paragraph [0029], where the user selects the

components to be employed and inter-connected and the system automatically creates data links between them). However, the combination of Molinari and Bowman is silent with respect to wherein updating the block diagram comprises displaying a wire connecting the first parameter value to the input terminal of the first node. On the other hand, Shah discloses **wherein updating the block diagram comprises displaying a wire connecting the first parameter value to the input terminal of the first node** (Shah, paragraph [0019], where a wire connecting the hardware device to the second component may be illustrated in the diagram). It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate Shah's teachings to the combination of Molinari and Bowman. A skilled artisan would have been motivated to do so, as suggested by Shah, paragraph [0010], in order to provide a system which aids a user in designing and implementing a measurement system. In addition, all of the references (Molinari, Bowman and Shah) disclose features that are directed to analogous art and they are directed to the same field of endeavor, such as creating software. This close relation between both of the references highly suggests an expectation of success.

Response to Arguments

6. Applicant's arguments with respect to claims 1-14, 20-27, 31 and 32 have been considered but are moot in view of the new ground(s) of rejection, necessitated by applicant's amendment filed on 12/19/2007.

Applicant argues that the combination of Molinari and Bowman does not teach "automatically include the first parameter value in the first function call in the source code of the software program in response to the user input selecting the first parameter value, wherein automatically including the first parameter value in the first function call comprises automatically updating the displayed source code to display the first parameter value within the first function call, wherein automatically including the first parameter value in the first function call aids a user in editing the first function call," as recited in newly amended claim 1, see the middle of page 13 to the top of page 14 of applicant's instant remarks.

Applicant is merely arguing the newly added limitations to claim 1, please see the rejection above.

Applicant argues that the combination of Molinari and Bowman does not teach "displaying a graphical user interface for selecting a parameter value for the first parameter of the first function call," as recited in newly amended claim 1, see the top of page 14 of applicant's instant remarks.

Applicant is merely arguing the newly added limitations to claim 1, please see the rejection above.

Applicant argues that Molinari teaches away from the combination with Bowman. Applicant further argues that Molinari teaches that an important advantage of the invention is that a measurement application created by a user of the invention may be fully represented in a brief text description of the chosen and configured aspects, and of their interconnections, in lieu of page of source code. The combination of Bowman with

Molinari would require the measurement application to be represented as source code, which would violate this principle of operation of Molinari's invention, see the middle of page 14 to the middle of page 15 of applicant's instant remarks.

Examiner respectfully disagrees.

Molinari does not teach away from the combination with Bowman. Molinari discloses an interface for the user to use in lieu of personally writing the source code. However, the program writes the underlying source code for the user. The program takes the user's inputs and turns it into source code in the form of the AIL file. Furthermore, paragraph [0038] of Molinari discloses "[a]s the user adds more panels, selects more properties, and effects connections between panels, the AIL file expands into a complete textual description of the user's program. At the same time, this AIL file corresponds to an *executable set of selections from existing libraries of executable code*. This AIL file, which is human-readable, and readily saved and stored, also constitutes an abstract representation of the existing, executable code segments that are invoked, ordered and activated upon running the user's program." In other words, the AIL file is turned into an executable (i.e., compiled) and the abstract representation of existing, executable code segments by the AIL file, is in essence, source code. Therefore, Molinari does not teach away from the combination with Bowman.

Applicant argues that the combination of Molinari and Bowman does not teach "display a block diagram of a graphical program, wherein the block diagram includes a plurality of interconnected nodes visually indicating functionality of the graphical program, wherein the block diagram can be compiled into executable code, wherein the

plurality of interconnected nodes includes a first node that takes a first input parameter," as recited in newly added claim 31, see the bottom of page 15 of applicant's instant remarks.

Applicant is merely arguing the newly added claim 31, please see the rejection above.

Applicant argues that the combination of Molinari and Bowman does not teach "automatically configure the first node with the first parameter value in response to the user input selecting the first parameter value, wherein automatically configuring the first node with the first parameter value comprises automatically updating the displayed block diagram to visually indicate that the first node receives the first parameter value as input," as recited in newly added claim 31, see the bottom of page 15 to the top of page 16 of applicant's instant remarks.

Applicant is merely arguing the newly added claim 31, please see the rejection above.

Conclusion

7. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Point of Contact

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hubert Cheung whose telephone number is (571) 270-1396. The examiner can normally be reached on M-R 7:30A - 5:00P EST; alt. F 7:30A - 4:00P EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached on 571-272-3642. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Tim T. Vo/
Supervisory Patent Examiner, Art Unit 2168

Examiner: Hubert Cheung
Date: February 16, 2008

/Hubert Cheung/
Examiner, Art Unit 2168